Sensors **2010**, *10*, 1021-1040; doi:10.3390/s100201021

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

CENet: A Cabinet Environmental Sensing Network

Zusheng Zhang [1,2,3,*], Fengqi Yu [2], Liang Chen [2] and Guangmin Cao [2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Road Zhongguancun, Haidian District Beijing, China

[2] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, 1068 Xueyuan Avenue, Shenzhen University Town, Nanshan District, Shenzhen, China; E-Mails: fq.yu@siat.ac.cn (F.Y.); liang.chen@siat.ac.cn (L.C.); gm.cao@siat.ac.cn (G.C.)

[3] Graduate University of Chinese Academy of Sciences, No.9 Yuquan Road, Sijingshan District Beijing, China

* Author to whom correspondence should be addressed; E-Mail: zushengzhang@gmail.com; Tel.: +86-755-863-92168; Fax: +86-755-863-92299.

Received: 29 November 2009; in revised form: 24 December 2009 / Accepted: 28 December 2009 / Published: 28 January 2010

Abstract: For data center cooling and intelligent substation systems, real time cabinet environmental monitoring is a strong requirement. Monitoring data, such as temperature, humidity, and noise, is important for operators to manage the facilities in cabinets. We here propose a sensing network, called CENet, which is energy efficient and reliable for cabinet environmental monitoring. CENet achieves above 93% reliable data yield and sends fewer beacons compared to periodic beaconing. It does so through a data-aided routing protocol. In addition, based on B-MAC, we propose a scheduling scheme to increase the lifetime of the network by reducing unnecessary message snooping and channel listening, thus it is more energy efficient than B-MAC. The performance of CENet is evaluated by simulations and experiments.

Keywords: wireless sensor network; cabinet monitoring; data-aided routing protocol; scheduling

---

1. Introduction

Cabinet environmental monitoring represents a typical application for wireless sensor networks [1,2]. From data center to substation, there are many cabinets and dedicated containers. These cabinets usually contain computers, high voltage switches, or other electronic equipments. Cabinet environmental monitoring is a technology that provides real-time recording of environment parameters in the interior of cabinets, such as temperature, humidity, smoking, noise, and so on. Depending on these data, operators can estimate the health and running status of the equipment in the cabinets. For example, a thermal alarm means a danger of potential hot spots. Operators need to identify the problem and make intelligent decisions on facility management.

A data center or substation can have several rooms. Each room can have dozens of cabinets. And every cabinet may have several racks. Operators need real-time monitoring of the equipment on each rack. In a substation, there are hundreds of cabinets which contain many kinds of electronic equipment, therefore huge numbers of points need to be monitored. Traditional solutions for cabinet environmental monitoring involve the use of wired sensors, connected through Ethernet interfaces [3]. However, those approaches suffer from high installation and configuration costs. Furthermore, sensor deployment is constrained by the availability of Ethernet connections.

Wireless sensor network (WSN) [4] is an ideal candidate for cabinet environmental monitoring because it is low cost and unattended technology. It can provide wide and dense coverage, and can be easily reprogrammed. Wireless sensors do not require network facility infrastructure in data centers or substations which is already a complicated environment with large amounts of equipment.

Cabinet monitoring introduces new challenges to wireless sensor networks. The network framework must have good scalability to support new equipment. Commands have to be reliably disseminated throughout the entire system so that all parts of the network can follow the commands to change their states. Sensed data must be reliably collected, e.g. temperature alarm information is too valuable to be lost due to communication errors or collisions. For battery powered wireless sensor networks, the network lifetime is a critical factor because replacing batteries for sensor nodes is a laborious task.

In this paper, we present CENet (Cabinet Environmental sensing Network). Focusing on network lifetime and reliability of data transmission, CENet provides two main contributions:

- CENet implements a data-aided routing protocol. Instead of using periodic beaconing to estimate link quality and maintain network routing topology, CENet uses transmitted data to continuously estimate link quality and its routing maintenance is triggered when poor link quality is detected. Therefore the energy dissipation due to beacons can be diminished. Compared to MintRoute's fixed 1 minute beacon interval [5], CENet sends 85% fewer beacons. It can also achieve a packets delivery ratio of more than 93%.
- CENet contains two kinds of nodes: master nodes and slave nodes. In order to prolong the network life time, we propose a scheduling scheme based on B-MAC [6] for slave nodes, which reduces unnecessary message snooping and channel listening, *i.e.*, a slave node keeps sleeping until it is time for sensing the environment or it has data for transmission. Using CENet, slave nodes consume about 65% less energy compared to using B-MAC, where the sample interval is 5 minutes.

The remainder of this paper is organized as follows. Section 2 gives the system architecture of cabinet environmental monitoring. Section 3 presents the design of routing protocol. Section 4 describes the scheduling scheme. Section 5 evaluates the performance of CENet by simulations and experiments. Section 6 describes related works. Finally, Section 7 gives a brief conclusion.

2. Architecture

CENet uses a hierarchical architecture, as shown in Figure 1. The gray circles denote slave nodes and the green circles denote master nodes. It has one base station (BS) which connects a computer through UART. We use Mica2 node as master and Dot node as slave, as shown in Figure 2. The node parameters are shown in Table 1. Slave nodes are placed inside cabinets and master nodes are typically installed outside of cabinets. Slave nodes are responsible for sensing data and transmitting packets to their master nodes. Each slave node can select only a master node as its parent. Master nodes form a routing tree and are responsible for forwarding data to the BS. They are not responsible for sensing the environment. In other words, slave nodes are the sensing front tier and master nodes are the communication infrastructure. The communication distances of slave nodes and master nodes are 0–3 meters and 0–15 meters, respectively. This short distance has the advantage of reducing communication interference because nodes are densely deployed.

The communication stack of CENet is as follows. A modified B-MAC [6] is used as MAC protocol. Mode 4 of B-MAC is used for master node scheduling, while our proposed scheduling based on B-MAC is used for slave node scheduling. The proposed data-aided routing protocol is used for both types of nodes. And the application layer contains two operations: data collection and event alarm.

Figure 1. CENet topology.

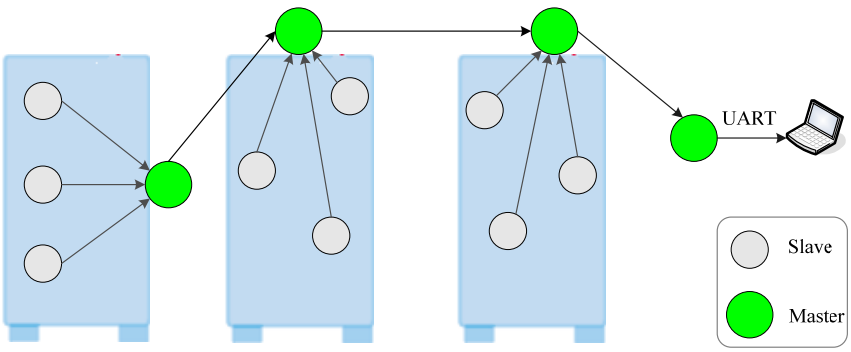


Figure 2. Mica2 node and Dot node.

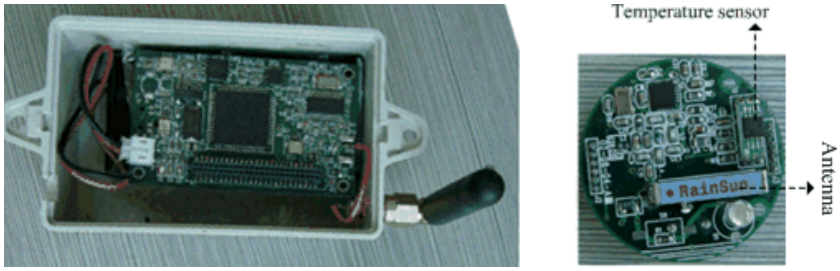


Table 1. Node parameters.

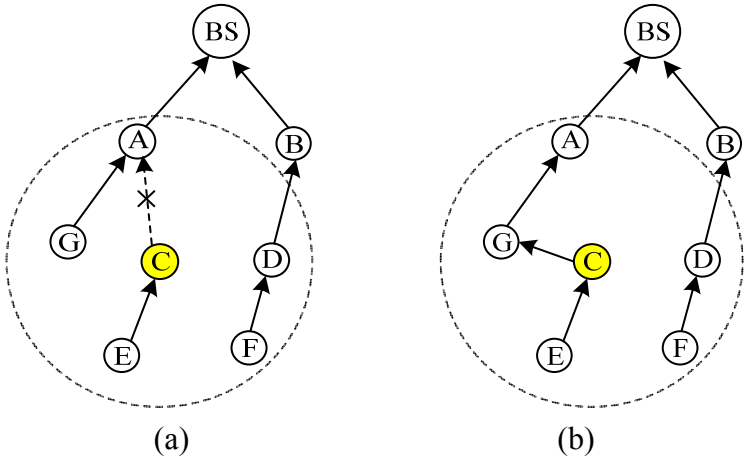
Mote Type	Dot	Mica2
Microcontroller	MSP430F1611	ATmega128
Radio	CC1101	CC1000
Data rate (kbps)	38.4	38.4
Power dissipation for reception (mA)	15.0	15.0
Power dissipation for transmission (0dBm) (mA)	16.0	16.0
Battery	A button battery	A pair of rechargeable batteries
Integrated sensors	Yes	No

3. Routing

Most existing routing protocols such as MintRoute [5], MultihopLQI [7], Directed Routing [8], Backbone Routing [9], etc. use periodic beaconing to estimate the link quality and maintain routing topology. Each node periodically measures the link qualities between itself and its neighbors by calculating the expected transmission (ETX [10]). The periodic beaconing has the following drawbacks:

(1) Periodic beaconing has a slow response to network variation. The link quality is a statistical result, so it must to be estimated after many beacon exchanges for more accuracy. In some situations, these beacon exchanges should be finished in a short time. For example, when a node parent has failed, the node needs to find a new parent right away. However, taking into account the energy efficiency, the protocols using periodic beaconing should not use short intervals. As shown in Figure 3, when the link from C to A is failed, C needs to select a new parent. Assume at least eight rounds of beaconing are needed to accurately estimate link quality. The default beaconing interval in MintRoute is 8 s and at least 64 s are needed to recover the failure route. CENet solves the problem by considering a network change as an event. When an event is detected, route maintenance is triggered. Many beacons are exchanged in a short time and failed routes can be quickly repaired.

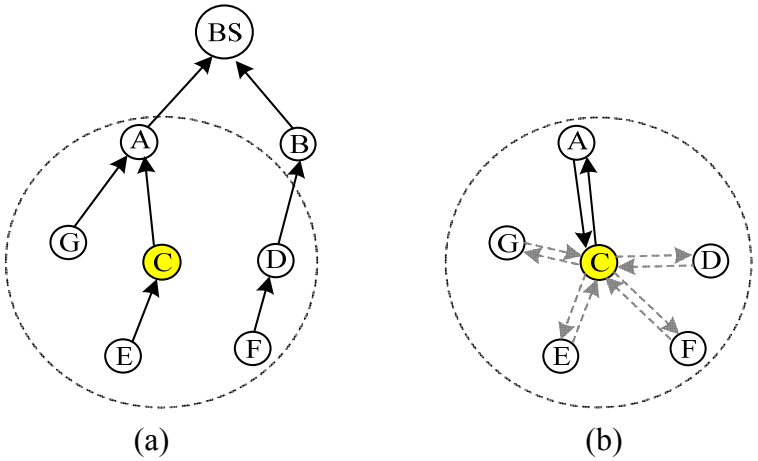
Figure 3. (a) The link from C to A is failed. (b) C selects G as its parent.



(2) Periodic beaconing is not energy efficient. A node periodically broadcasts beacons to its neighbors without considering link quality and data transmission. We denote the link from a node to its

parent as L2P. A node selects the neighbor node with the best link as its parent. When L2P quality is good enough to support data transmission reliability, it is not necessary to change parent. As shown in Figure 4 (a), the parent of node C is A and only the bidirectional link between C and A effects their communication reliability. Let's focus on the nodes inside the circle, as shown in Figure 4 (b). Node C broadcasts beacons to its neighbors and estimates the link qualities periodically. But for the link between C and A, the beacon exchanges between node G, D, E and F are redundant and waste energy. CENet solves the problem as follows: when a node is trying to find a parent, the link qualities of all the routes connected to the BS are estimated using beacons. When the node has found its parent, it stops issuing beacons. The link quality variation of L2P is monitored by transmitted data.

Figure 4. (a) A routing topology. (b) Focus on the nodes in the circle of (a). For node C, only the bidirectional link between C and A effects the reliability of packet transmission.



Our data-aided routing protocol can overcome the shortcomings of the current periodic beaconing routing protocols. It is more energy efficient and reliable for data packet transmission.

3.1. Routing Tree Construction

In CENet, beacons are divided into three categories: request, reply and pull. All of these beacons are transmitted by broadcasting. A request beacon is to advertise that the sender doesn't have a proper parent and it is searching for a parent. A reply beacon is to indicate that the sender has a route to BS and can be a candidate parent. Finally, a pull beacon indicates that the sender has better or updated route information and notifies its neighbors to maintain their routes.

When a node does not have a parent or its parent turns to fail, it starts searching for a parent by periodically broadcasting request beacons to its neighbors. A neighbor node with a determined parent broadcasts a reply beacon when it receives the request. The request and reply beacons need to exchange many times to accurately estimate the link qualities. We use expected transmission (ETX [10]) as a cost metric, which reflects the quality of the link to one neighbor. The more reliable the link quality is, the smaller the ETX. The route cost is the sum of link ETX values along the entire route. A node selects a parent with the least route cost. Once a node has found its parent, it stops sending requests.

The frequency of sending request beacons is determined by user requirements. For cabinet monitoring application, if a user wants to construct a routing tree quickly after sensor deployment, they can set the interval to several hundreds of milliseconds.

After sensor deployment, only BS has a parent which is an upper computer connected over UART. Other nodes do not have parents and broadcast requests periodically. When BS receives the requests from the nodes closer to it, it sends reply beacons. After enough exchanges of the beacons, the sensor nodes at the 1st level in the routing tree choose BS as parent and stop sending requests. Similarly the nodes at the 2nd level exchange request and reply beacons with the nodes at the 1st level to determine their parent. The whole routing tree can be constructed after several rounds. When the routing tree is constructed, none of the nodes in the network sends request beacons. If the communication links in the network remain good, there will be no beacon exchange at all and the link quality is estimated by transmitted data.

3.2. Link Quality Estimation

Since CENet uses acknowledge and retransmission schemes, the decline of link quality increases the number of retransmissions and packet drops. The link quality is estimated by observing the number of packet retransmissions. CENet uses the window mean with EWMA (exponentially weighted moving average) as link quality estimator [11] which computes an average retransmission rate over a time period and smoothes the average by EWMA.

Let i be the index of a time window, and t be the time window represented in the number of message transmissions and C_i be the number of packet retransmissions in the i^{th} time window. H_i is a value between C_i and history retransmissions, H_i can be expressed as:

$$H_i = \begin{cases} 0, & i = 0 \\ C_i, & i > 0 \text{ and } H_{i-1} = 0 \\ H_{i-1} * a + (1-a) * C_i, & i > 0 \text{ and } H_{i-1} > 0 \end{cases} \quad (1)$$

$$T_i = \lceil H_i * (1+b) \rceil \quad (2)$$

where $0 < a < 1$ and $0 < b < 1$ both are tuning parameters, T_i is a threshold. In the time window t, if H_i is greater than T_{i-1}, the route maintenance is triggered and H_i will be reset to 0 in next round estimation. To understand (1) and (2), we give an example, let C_i be a random value between 7 and 10, $a = 0.5$ and $b = 0.1$. Table 2 shows the retransmission result of five time windows. In the 3rd and 5th windows, the number of retransmissions is greater than the threshold and routing maintenance is triggered.

From Table 2, we can see a problem in the link estimator. When $T_{i-1} = 0$, C_i has a useless reference value and the estimator will never be able to detect the decrease of link quality in the current time window. To solve the problem, we set two criteria for routing maintenance. One is the estimator mentioned above. The other is data packet drop rate. For example, when the packet drop rate is greater than 10% (according to the reliability level of the user requirement) in a time window, the routing maintenance process will be triggered.

Table 2. Example of estimator.

i	**1**	**2**	**3**	**4**	**5**
C_i	8	7	10	7	9
H_i	8	7.5	0	7	0
T_i	9	9	0	8	0
Routing maintenance	No	No	Yes $C_3 > T_2$	No	Yes $C_5 > T_4$

3.3. Routing Tree Maintenance

In our data-aided routing protocol, the route maintenance of a node is triggered under the following events:

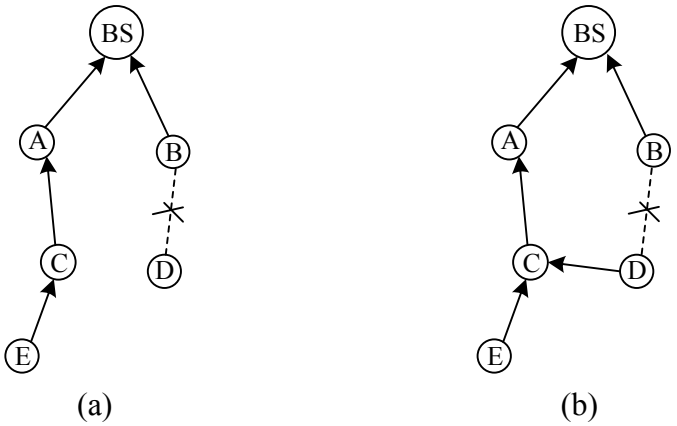
(a) When a node's link quality to its parent gets worse or the link has failed.

(b) When a node receives a pull beacon.

After routing maintenance has been triggered, a node sends request beacons periodically to select a proper parent. If it finds a parent and the route cost to BS is significantly decreased (we choose 20% decrease as threshold), the node broadcasts this event using pull beacons because it may provide a better route for nearby nodes. On the other hand, if the new route cost is significantly increased (we choose 20% increase as threshold) or it cannot find a parent, the node also informs the event to its children using pull beacons so that they will adjust the network topology in time.

The scene of route failure is demonstrated in Figure 5 (a). The link quality between node D and its parent B is estimated to be worse by transmitted data packets, and routing maintenance is triggered. As shown in Figure 5 (b), C is selected as its new parent.

Figure 5. Declining of link quality triggers routing maintenance.



As shown in Figure 6 (a), when node C detects that node A is removed or failed, it can't find an available parent in its communication range. C tells its children E and F that the route is invalid by broadcasting pull beacons. Routing maintenance of E and F is triggered and a new routing tree is reconstructed as shown in Figure 6 (b).

When a new node joins network, routing maintenance is triggered. As shown in Figure 7 (a), node F is a new node and its parent is BS. F broadcasts pull beacons. D receives the pull beacons and routing maintenance process is triggered. Eventually D chooses F as its new parent, as shown in Figure 7 (b).

If the change in link cost is significant (20%), D will inform its neighbors the cost change by pull beacons.

Figure 6. Parent node informs its children to trigger routing maintenance.

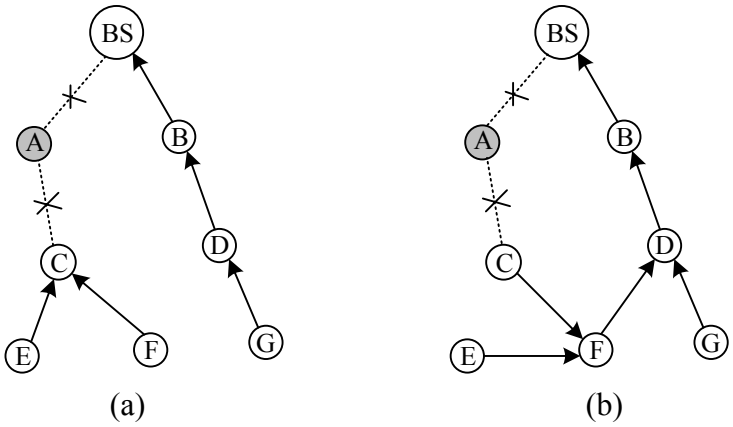
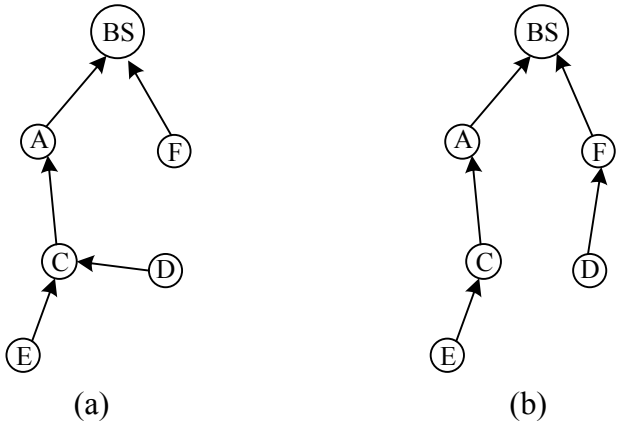


Figure 7. When a new node F joins network, routing maintenance is triggered.



4. Scheduling

Radio communication dominates the power consumption in wireless sensor networks [12]. Sensor nodes waste a lot of energy for receiving signals. The energy consumption for receiving is approximately equal to that for data transmission. In addition, a node in idle mode also consumes energy. Therefore, idle listening, the time spent for waiting packets, consumes a significant amount of energy. Stemm *et al.* [13] observed that idle listening dominated the energy consumption of network interfaces in hand-held devices. It becomes clear that using scheduling to reduce listening and turn off the radio during idle time is an effective way to prolong network lifetime.

4.1. Modified Scheduling Scheme Based on B-MAC

There are two kinds of scheduling: (1) Integrated scheduling with MAC layer, such as B-MAC [6], S-MAC [14]; (2) Independent scheduling based on routing topology, such as FPS [15], DB-TDMA [16], and DCS [17]. Independent scheduling protocols need time synchronization and energy consuming.

CENet adopts B-MAC which has friendly interfaces that allow users to flexibly control the operation of scheduling. It has better performance than S-MAC.

For hierarchical architecture of cabinet monitoring, a master node can't forecast when to receive data packets from its children. In B-MAC, to ensure sensed data being forwarded to the BS reliably, master nodes must detect data when each of its neighbors transmits a packet, regardless of the packet destination. In CENet, mode 4 of B-MAC is selected as the scheduling of master node, which is the optimum mode of B-MAC. Master nodes wake up every 100 ms to detect channel activity. On the other hand, slave nodes, which do not forward data, can decide when to send data. Based on B-MAC, we have developed a scheduling scheme for slave nodes.

Figure 8. The pseudo code of the proposed scheduling.

```
// For slave node
// T is the sampling interval
1    If (no master)
2            scheduling uses mode 4 of B-MAC
3    If (have master)
4            set the interval of wake up timer to be T
5            go to sleep
6    At timer T
7            wake up
8            scheduling uses mode 4 of B-MAC
9            sensing environment
10           piggyback the parameter version number on data packet
11           send data packets to parent node
12   Case: receive acknowledge from parent
13           if (ACKFlag= 0) // ACKFlag =0 indicates the parameter version is the latest
14                   set the wake up timer to be T
15                   go to sleep
16   Case: receive updated parameters from master
17           send acknowledge
18           go to sleep
19   Case: don't receive the acknowledge
20           resend data packet
21   Case: the number of retransmissions is over a threshold
22           set current parent to null
23           go to line 1

// For master node
1    Case: receive data from its slave node
2            check parameter version number
3            if (the same)
4                    send ACKFlag = 0 //ACKFlag=0 indicates the parameter version is the latest
5            else if (different)
6                    send ACKFlag = 1 //ACKFlag=1 indicates the parameter version is old
7                    send updated parameters to its slave node
```

The proposed scheduling must support beacon and command transmission. Figure 8 describes the pseudo code of the scheduling. Each node has two states: sleep and active. When a slave node needs to communicate with its master, it enters active state. Because communication nodes must be in the same mode in B-MAC, the scheduling for active state is mode 4 of B-MAC. A slave node wakes up at

predetermined time to sense the environment and sends data to its master. Meanwhile, it piggybacks the parameter version number on data packet to check whether the parameters are the latest. If its master has a new version of parameters, it sends updated parameters to its slave immediately. If the number of data packet retransmissions or drop rate is over a threshold, the slave node starts to look for a new master. Otherwise the slave node goes to sleep.

4.2. Analysis of Energy Consumption

In this section, we compare the energy consumption between mode 4 of B-MAC and our modified scheduling scheme. Ignoring the energy consumption in parameter update and beacon exchange, the energy consumption of a slave node consists of energy consumed by transmission (E_{tx}), data sensing (E_s), receiving (E_{rx}), listening (E_{listen}), and sleep (E_{sleep}). The related parameters [6] are shown in Tables 3 and 4, where the average listening current and listening time are obtained in our experiment which will be discussed in detail in the second paragraph of Section 5.2.2. Taking the sample interval as 5 minutes, we can calculate the energy consumption of a slave node in a sample interval, denoted as E

Table 3. Energy consumption.

Operation	**Energy (*mJ*)**		**Time (s)**	
Receive a packet	5.74704	E_{rx}	0.127	t_{rx}
Transmit a packet	7.66272	E_{tx}	0.127	t_{tx}
Sense environment	66.0	E_s	1.1	t_s

Table 4. Related parameters.

Notation	**Parameter**	**Value**
I_{sleep}	Sleep current (*mA*)	0.03
I_{listen}	Average listening current (*mA*)	2.0
V	Voltage	3.0
t_{listen}	Listening time (s)	0.01
n	Number of neighbors	3

$$E = E_{tx} + E_s + nE_{rx} + E_{listen} + E_{sleep} \tag{3}$$

$$E_{listen} = I_{listen} T_{listen} V \tag{4}$$

$$T_{listen} = t_{listen} N_{listen} \tag{5}$$

$$E_{sleep} = I_{sleep} T_{sleep} V \tag{6}$$

$$T_{sleep} = T - t_{tx} - t_s - nt_{rx} - T_{listen} \tag{7}$$

where n is the number of neighbors; T is the sample interval and it is equal to 5 min × 60 = 300 s; N_{listen} is the number of listening times in T; T_{listen} is the total listening time in T; T_{sleep} is the total sleep time in T.

Using mode 4 of B-MAC, the default value of channel listening interval is 0.1 s, and the period of a listening is t_{listen} = 0.01 s. In five minutes, N_{listen} = 5 min × 60/ (0.1 s + 0.01 s) = 2,727 and T_{listen} = 27.27 s, so E_{listen} = 163.6 *mJ*. In B-MAC, a node can detect and receive data packet transmitted from its neighbors. If each of its neighbors sends a data packet, the node will receive *n* packets. Given *n* = 3, the energy consumption for receiving the *n* packets is 17.24 *mJ*. T_{sleep} can be calculated as T_{sleep} = 271.12 *s* and E_{sleep} = 24.40 *mJ*, so *E* = 278.92 *mJ*. Using mode 4 of B-MAC, the energy consumption for listening and data receiving takes 58.6% and 6.2% of the total energy dissipation, respectively.

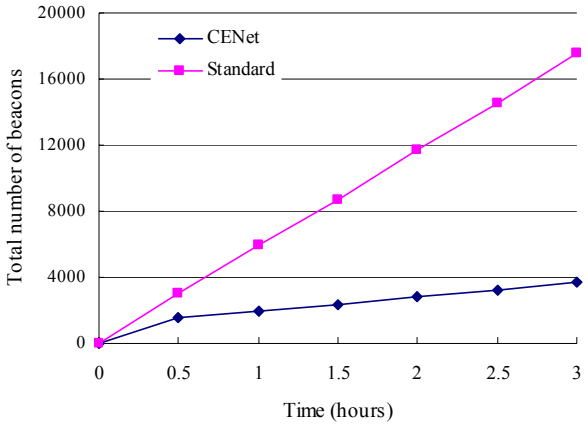
While using the modified scheduling scheme, a slave node does not need to receive data packets, neither listening to the channel. The energy consumption of a slave node only consists of the energy consumption for data sampling (E_s), data transmission (E_{tx}), and sleep (E_{sleep}). We can calculate T_{sleep} = 298.873 s, E_{sleep} = 26.898 *mJ*, and *E* = 100.5613 *mJ*. Therefore CENet consumes about 65% less energy compared to mode 4 of B-MAC when the sample interval is 5 minutes.

5. Evaluation

5.1. Simulations

In this section, CENet is evaluated in simulations. A prototype was implemented in TinyOS [18]. CENet system is compared to standard TinyOS networking stack (MintRoute [5] on top of B-MAC [6]), standard stack for short, in terms of reliability and energy efficiency. TOSSIM [19] is used in all simulations, which is a discrete event-based simulator for TinyOS. TOSSIM requires a user specify the signal attenuation levels for every network link. We used the Log Distance Path Loss model to calculate these attenuations [20]. One hundred and one nodes are deployed in an area of (50 m, 50 m). There were 70 slave nodes, 30 master nodes, and one BS placed at (0, 0).

Figure 9. The number of beacons for CENet decreases after a period of time. It is significantly less than that of standard stack.

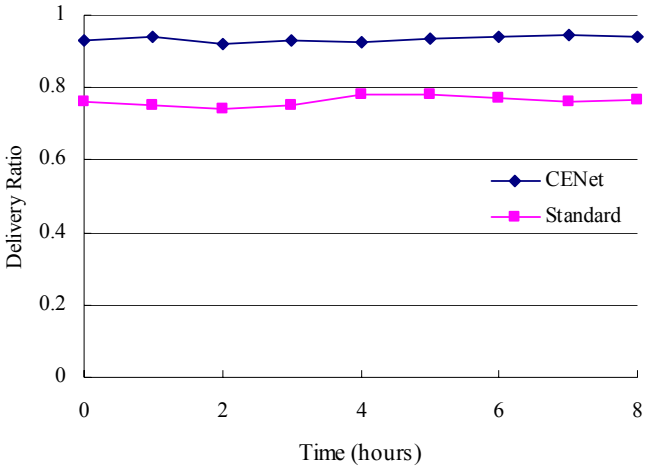


The number of beacons for CENet and standard stack are shown in Figure 9. The beaconing rate of CENet is high during network startup because every node is busy with exchanging beacons to discover the routes and construct the network. The beaconing rate decreases after a period of time, because in

CENet beacons are only sent when the route needs be maintained. In MintRoute, beacons are sent at a fixed interval of 1 minute, which is a standard stack routing protocol. CENet has much less number of control packet transmissions than MintRoute.

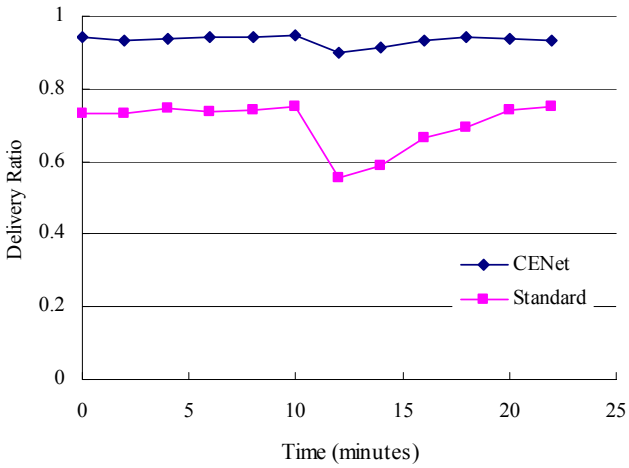
Figure 10 shows the simulation results when CENet and standard stack are run for over 8 hours, respectively. The maximum number of retransmissions for CENet and standard stack are both 4. The delivery ratio for CENet and standard stack are 93% and 76%, respectively. Therefore CENet has higher reliability than the standard stack.

Figure 10. CENet has a consistently higher delivery ratio than standard stack.



The ability of route failure recovery is studied by running CENet and the standard stack for 25 minutes with a packet transmission interval of 10 s. After 10 minutes, we remove five master nodes that are forwarding most of the packets in the network. The simulation results are shown in Figure 11. It shows that CENet has only a small drop in delivery ratio at 12 min. This is because when a node detects a route failure, it transmits beacons periodically with an interval of several hundreds milliseconds. Thus the failure route can be repaired quickly. However, the delivery ratio for standard stack is 55%. This is because its routing protocol is MintRoute that sends beacons every 1 minute. Therefore it needs several minutes to repair the failure.

Figure 11. Robustness of CENet and standard stack when five master nodes are removed in the 10th minute.



The energy consumption for CENet and standard stack is studied by the following simulations. Slave nodes generate sensing data every 30 seconds and MintRoute beaconing interval is 1 minute. Figure 12 shows that the average node energy consumption is 9.1 mAh for the standard stack and 5.65 mAh for CENet over 3 hours. Therefore CENet is about 38% more energy efficient than the standard stack. Figure 13 compares the energy consumption of delivering each data packet from source node to BS for CENet and standard stack. It shows that control packets for CENet are much less than those for the standard stack (3% *vs.* 7.2%). The decrease in the number of control packets is due to the data-aided routing protocol, where beacon exchange happens only when route needs to repair.

Figure 12. Energy consumption of CENet and standard stack.

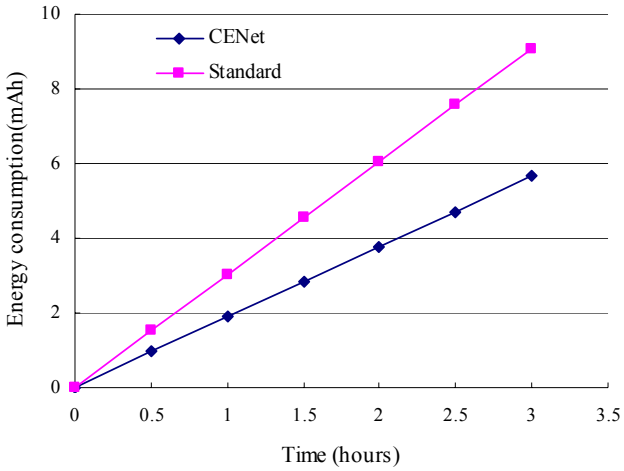
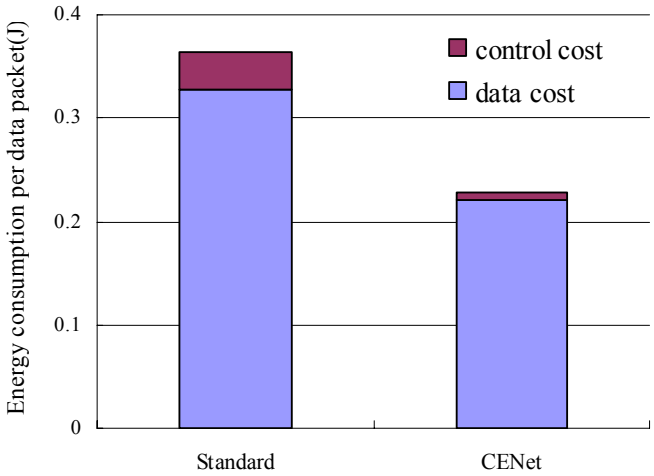


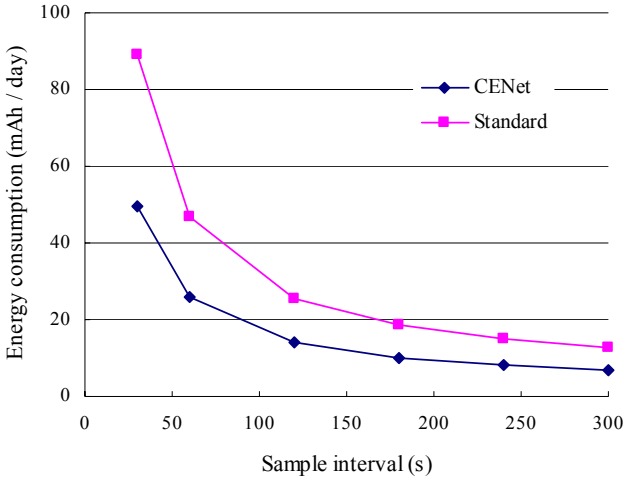
Figure 13. The energy consumption of CENet is 38% less than that of standard stack and the energy consumption for control packets in CENet is 85% less than that in standard stack.



CENet and standard stack are evaluated in terms of overall network lifetime. Considering the tradeoff between energy consumption and response time to network change, let the beacon interval of MintRoute be equal to the sample interval of slave nodes, which varies from 30 s to 300 s. As shown in Figure 14, when the sample interval is 30 s, the average energy consumption of slave node is 49 mAh/day for CENet and 86 mAh/day for standard stack. So a pair of AA batteries (2,500 mAh) can

serve 51 days for a node running CENet and 29 days for running the standard stack. When the sample interval is 5 minutes, the average energy consumption of each node is 6.9 mAh/day for CENet and 12.8 mAh/day for standard stack. So the network lifetime can be longer than one year for a CENet network.

Figure 14. Average energy consumption of CENet and standard stack for sample interval between 30 s and 300 s.



5.2 Experiments

5.2.1. Cabinet Environmental Deployment

CENet monitors the temperature of the isolated contacts in high voltage cabinets. The cabinet is steel and has a plastic window for the operator to see the equipment running state, so the RF signal can transmit through the window. Slave nodes are deployed in the cabinets for temperature sensing. Master nodes are placed outside the cabinet.

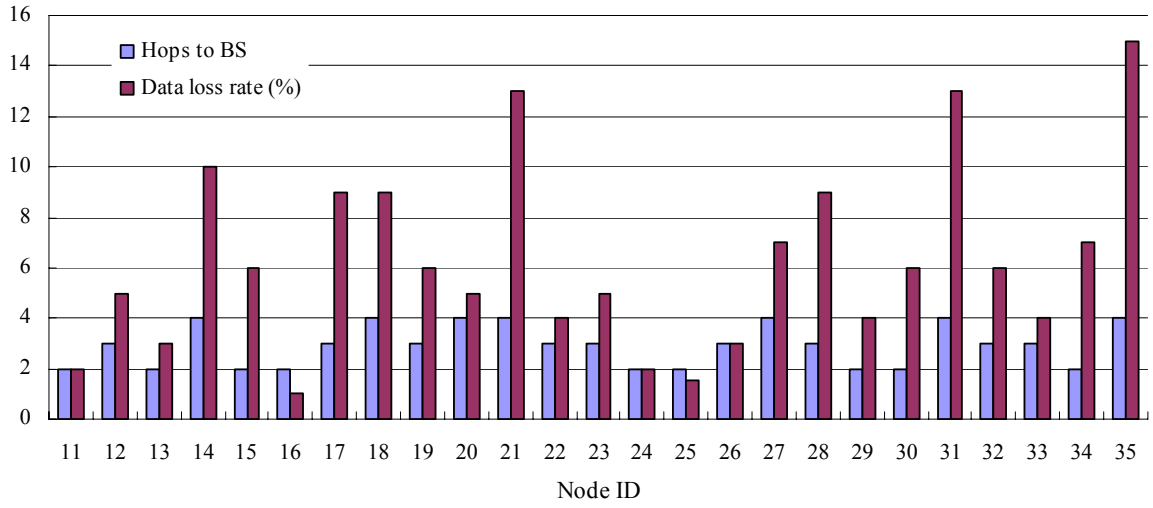
Thirty six nodes are deployed in high voltage cabinets in a substation. There are one BS node, 10 master nodes, and 25 slave nodes. The BS connects to the computer through UART. The nodes with ID numbers from 1 to 10 are master nodes, and 11 to 35 are slave nodes. The slave node mounts a DS75 temperature sensor. It senses environment temperature and transmits the data every 5 minutes. Master nodes are not equipped with any sensors and are only responsible for data packet forwarding. We set the request beacon interval to be 0.5 s. The RF transmission power for slave node is −20 dBm, and for a master node it is −10 dBm. CENet drops a data packet after four retransmissions, which can be increased by the user to have a more reliable data yield.

The routing tree construction is completed within 30 seconds. The maximum number of hops of a routing tree is 4. Two criteria are adopted to estimate the link quality. The first one is the number of retransmissions, as described in Section 3.2.1, where the time window size is 12 data transmissions, $a = 0.5$, and $b = 0.2$. The second one is packet drop rate. The threshold of the packet drop rate is 10% in a time window.

The network was running for 17 days. The BS received 115,056 data packets which are stored in a PostgreSQL [21] database. Because every data packet includes an attribute of sequence number, BS

can calculate packet loss rate, as shown in Figure 15. The data packet loss rate varies between 0% and 15% with an average of 5.6%. From Figure 15, one can see that nodes with more hops have greater packet loss rate. This is because the retransmission mechanism in CENet only provides a certain reliability of transmission at each hop (hop-by-hop reliability). In other words, it can't ensure end-to-end reliable packet delivery.

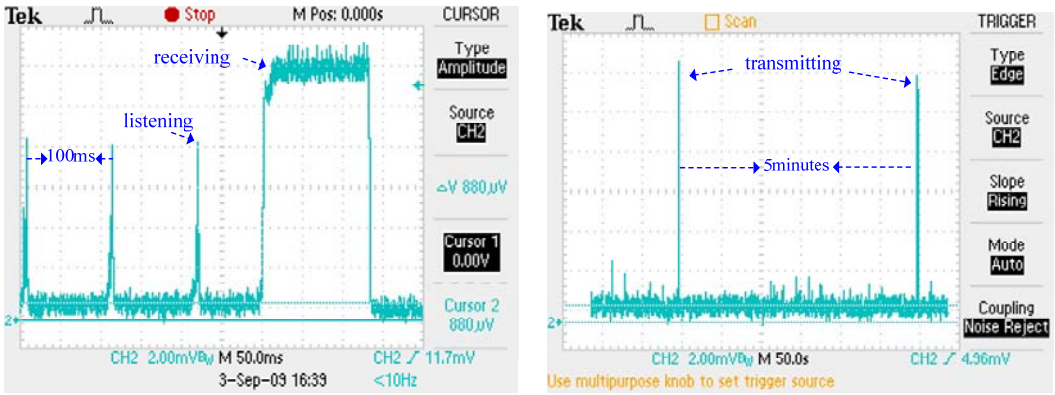
Figure 15. The average packet loss rate is 5.6% and nodes with more hops to BS have greater packet loss rate.



5.2.2. Energy Consumption

The energy consumption of master and slave node is measured by a Tektronix-2024B oscilloscope. Figure 16 illustrates the energy consumption of master and slave node. One can clearly see that master node listens to channel every 100 ms. On the other hand, the slave node sleeps most of the time and only wakes up for data packet transmission.

Figure 16. (a) Power consumption of master node. (b) Power consumption of slave node.

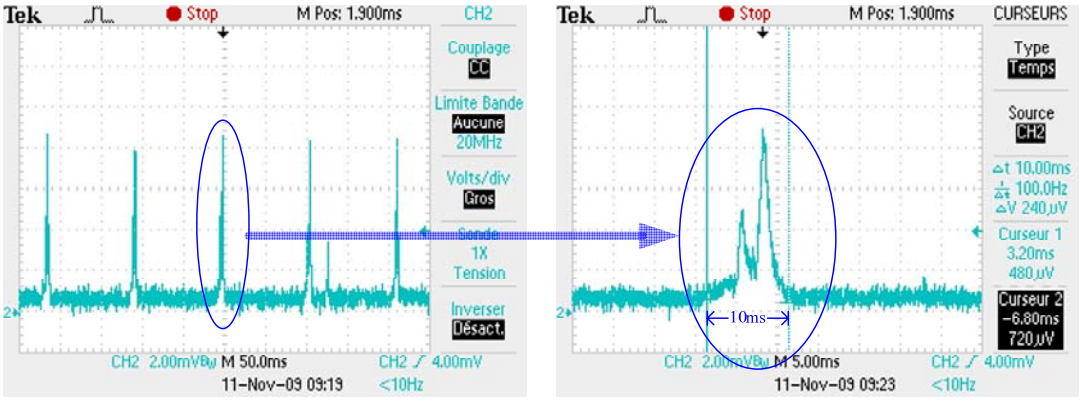


(a) (b)

Using mode 4 of B-MAC, the average listening current of a slave node is measured. As shown in Figure 17, a node listens to the channel for 10 ms with an interval of 100 ms. Figure 18 shows the

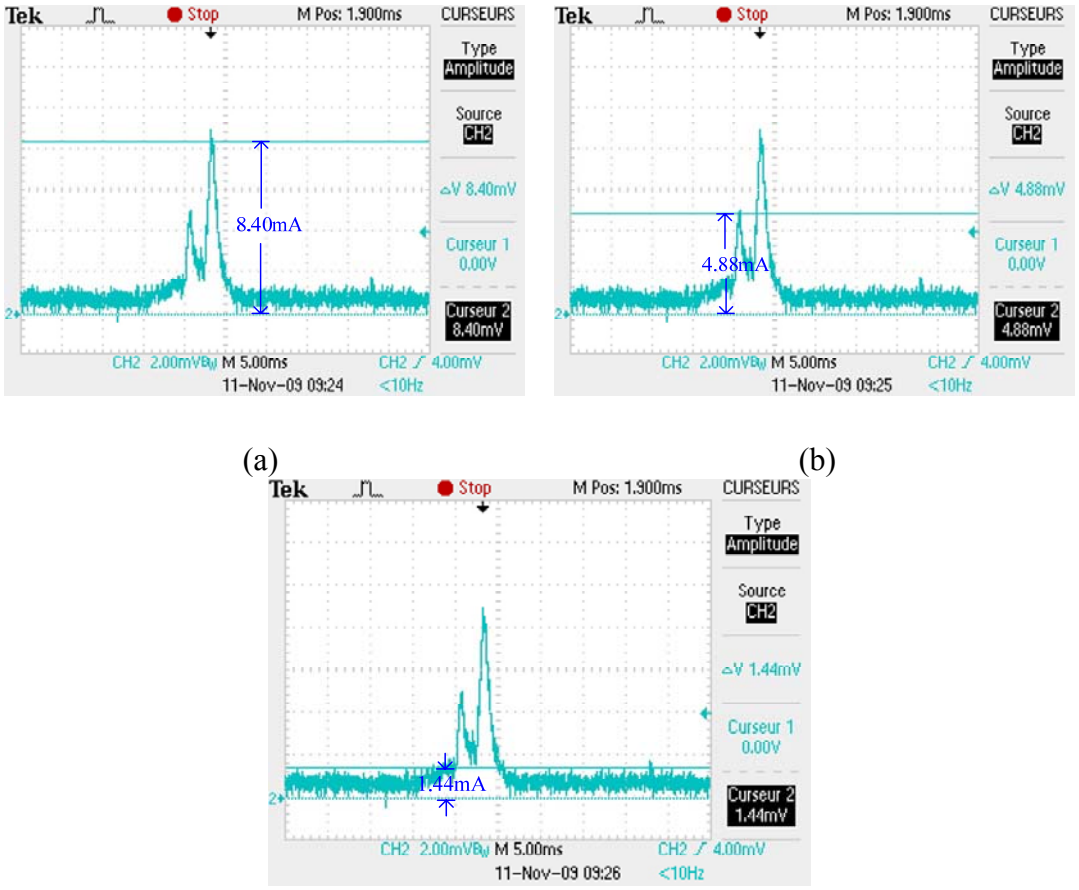
measured current values. The average listening current is calculated as 2.327 mA. Considering the environment noise, we set the average listening current to be 2 mA, which does not have any effect on our analysis.

Figure 17. (a) Node listens to the channel for 10 ms with an interval of 100 ms. (b) The measure current in one listening period.



(a) (b)

Figure 18. The detailed values of the measured listening current.



(a) (b)

(c)

5.2.3. The Effect of Interference

To test how the interference affects data packet transmission reliability, we have done a series of experiments. A sensor network contains one BS, three master nodes and five slave nodes. The RF transmission power for each node is −20 dBm, and the sample interval for slave node is 30 s. Firstly, we placed the network in a sealed steel box. The external interference can be ignored. The link quality is good because they are close to each other [5]. The average delivery ratio of CENet and standard stack (standard TinyOS networking stack) are both 100%. Secondly, we placed the network in a laboratory and a three-hop tree is constructed. As shown in Table 5, the average delivery ratio of CENet is 94.5% and for standard stack it is 82.3%.

Finally, an interference node is introduced into the sensor network and its MAC protocol is modified by disabling the backoff mechanism. It continuously broadcasts data and does not consider the channel status (busy or idle). The interference is placed into the network. As shown in Table 5, CENet has delivery ratio of 89.2%. However, the delivery ratio of the standard stack is decreased to 64.7%. That is mainly because the MintRoute routing protocol cannot quickly repair an interfered route while CENet is robust to link interference.

Table 5. The average delivery ratio.

	Sealed steel box	**Laboratory**	**With interference**
CENet	100%	94.5%	89.2%
Standard	100%	82.3%	64.7%

6. Related Works

Many works have been done for environmental monitoring, for example, health monitoring of mechanical machines [22], volcano monitoring [23], earthquake monitoring [24], redwood monitoring [25], underground coal mine monitoring [26], structural health monitoring [27,28], and habitat monitoring [29]. However, current networks can't satisfy the requirement of cabinet environmental monitoring which requires dense deployment and long network lifetimes.

Nasipuri *et al.* presented a wireless sensor network that is deployed in a substation for monitoring the health of power subsystems, such as circuit breakers, transformers, and transformer bushings [30]. They used a mixture of solar-powered nodes and battery-powered nodes to construct a hierarchical topology. However it is an outdoor deployment sensing network and can't be adopted for cabinet environmental monitoring.

Genomote [2] is a project for data center monitoring, which is developed at Microsoft Research, where IEEE 802.15.4 wireless technology over wired networks and WiFi is adopted. Two classes of Genomotes are designed: master motes and slave sensors. A slave sensor connects with a master mote using a wired medium. Because of the partly wired connections, it is inconvenient for sensor deployment. In this project, a reliable data acquisition system [31] is developed and it can achieve 99% reliable data yield. Master motes are powered by computer through USB connection. Energy

efficiency is not considered in the data acquisition system and thus it can't be used in battery-powered wireless sensor networks.

7. Conclusions and Future Work

Cabinet environmental monitoring represents a typical application for wireless sensor networks. We have proposed CENet for cabinet environmental monitoring. CENet uses two kinds of nodes: slave and master, to construct a hierarchical architecture. The slave nodes are the front sensing tier and the master nodes are the communication infrastructure which is responsible for forwarding data from slaves to BS. CENet implements a data-aided routing protocol which achieves 93% delivery ratio with fewer beacon transmissions compared to periodic beaconing. In addition, using our proposed scheduling, a slave node stays in sleep mode in most of the time and only wakes up for sensing data transmission, which helps prolong network lifetime.

A CENet may consist of hundreds of nodes within high voltage cabinets in a substation. The communication interference can be serious. One way to increase data throughput and reduce data latency is using multiple BSs at different locations. Furthermore, we can take advantage of multi-channel sensor network to reduce interference. As future work, we intend to improve CENet to support multiple BSs and channel diversity.

Acknowledgements

This work was supported by the National S&T Major Project of China under Grant No. 2009ZX03006-001.

References and Notes

1. Punchimudiyanse, M. Environment Monitoring System for Distributed Network Cabinets in Typical Campus Network. In *Proceedings of the International Conference on Industrial and Information Systems (ICIIS)*, Nawala, Sri Lanka, August 2007; pp. 17-20.
2. Liu, J.; Zhao, F.; Reilly, J.; Souarez, A.; Manos, M.; Liang, C.; Terzis, A. Project Genome: Wireless Sensor Network for Data Center Cooling. *Architect. J.* **2009**, *18*, 28-34.
3. Patel, C.D.; Bash, C.E.; Sharma, R.; Beitelmal, M.; Friedrich, R. Smart Cooling of Data Centers. In *Proceedings of International Electronic Packaging Technical Conference and Exhibition*, Maui, HI, USA, June 2003.
4. Akyildiz, I.F.; Su, W.L.; Sankarasubramaniam, Y.; Cayirci, E. A Survey on Sensor Networks. *IEEE Commun. Mag.* **2002**, *8*, 102-114.
5. Woo, A.; Tong, T.; Culler, D. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, CA, USA, November 2003; pp. 14-27.
6. Polastre, J.; Hill, J.; Culler, D. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Baltimore, MD, USA, 2004; pp. 95-107.

7. *The MultiHopLQI protocol*. Available online: http://www.tinyos.net/tinyos-2.x/tos/lib/net/lqi/ (accessed on September 12, 2009).
8. Maroti, M. Directed Flood-Routing Framework for Wireless Sensor Networks. In *Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware*, Toronto, Canada, October 2004.
9. He, T.; Krishnamurthy, S.; Stankovic, J.; Abdelzaher, T.; Luo, L.; Stoleru, R.; Yan, T.; Gu, L.; Hui, J.; Krogh, B. An Energy-efficient Surveillance System Using Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys)*, Boston, MA, USA, 2004; pp. 270-283.
10. De Couto, D.S.J.; Aguayo, D.; Bicket, J.; Morris, R. A High-throughput Path Metric for Multi-hop Wireless Routing. In *Proceedings of the International Conference on Mobile computing and networking (MobiCom)*, San Diego, CA, USA, September 2003; pp.134-146.
11. Woo, A.; Culler, D. *Evaluation of Efficient Link Reliability Estimators for Low-power Wireless Networks. Technical Report UCB//CSD-03-1270*, U.C. Berkeley Computer Science Division: Berkeley, CA, USA, September 2003.
12. Doherty, L.; Warneke, B.A.; Boser, B.E.; Pister, K. Energy and Performance Considerations for Smart Dust. *Int. J. Par. Distr. Sys. Netw.* **2001**, *4*, 121-133.
13. Stemm, M.; Katz, R. Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices. *IEICE Trans. Commun.* **1997**, *8*, 1125-1131.
14. Ye, W.; Heidemann, J.; Estrin, D. An Energy-efficient MAC Protocol for Wireless Sensor Network. In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, San Francisco, CA, USA, 2002; pp. 23-27.
15. Hohlt, B.; Doherty, L.; Brewer, E. Flexible Power Scheduling for Sensor Networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*, Berkeley, CA, USA, 2004; pp. 205-214.
16. Zhang, Z.; Yu, F.A Depth-based TDMA Scheduling for Clustering Sensor Networks. In *Proceedings of the 4th International Conference on Frontier of Computer Science and Technology*, Shanghai, China, December 2009.
17. Wu, H.; Luo, Q.; Xue, W. Distributed Cross-Layer Scheduling for In-Network Sensor Query Processing. In *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM)*, Pisa, Italy, March 2006; pp. 176-189.
18. Levis, P.; Madden, S.; Polastre, J.; Szewczyk, R.; Whitehouse, K.; Woo, A.; Gay, D.; Hill, J.; Welsh, M.; Brewer, E.; Culler, D. TinyOS: An Operating System for Sensor Networks. In *Ambient Intelligence*; Weber, W.; Rabaey, J.; Aarts, E., Eds; Springer-Verlag: Berlin, Germany, 2005, pp. 115-148.
19. Levis, P.; Lee, N.; Welsh, M.; Culler, D. Tossim: Accurate and Scalable Simulation of Entire TinyOS Applications. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, CA, USA, 2003; pp. 126-137.
20. Rappaport, T.S. *Wireless Communications: Principles and Practices*, 2nd ed.; Prentice Hall Press: Englewood Cliffs, NJ, USA, 2001.
21. Stonebraker, M.; Kemnitz, G. The POSTGRES Next Generation Database Management System. *Commun. ACM*. **1991**, *34*, 78-92.

22. Krishnamurthy, L.; Adler, R.; Buonadonna, P.; Chhabra, J.; Flanigan, M.; Kushalnagar, N.; Nachman, L.; Yarvis, M. Design and Deployment of Industrial Sensor Networks: Experiences from A Semiconductor Plant and the North Sea. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, USA, November 2005; pp. 64-75.
23. Werner-Allen, G.; Johnson, J.; Ruiz, M.; Lees, J.; Welsh, M. Monitoring Volcanic Eruptions with A Wireless Sensor Network. In *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN)*, Istanbul, Turkey, January 2005; pp. 108-120.
24. Glaser, S.D.; Chen, M.; Oberheim, T.E. Terra-Scope™ A MEMA-based Vertical Seismic Array. *Smart Struct. Syst.* **2006**, *2*, 115-126.
25. Tolle, G.; Polastre, J.; Szewczyk, R.; Turner, N.; Tu, K.; Buonadonna, P.; Burgess, S.; Gay, D.; Hong, W.; Dawson, T.; Culler, D. A Macroscope in the Redwoods. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, USA, November 2005; pp. 51-63.
26. Li, M.; Liu, Y. Underground Coal Mine Monitoring with Wireless Sensor Networks. *ACM Trans. Sensor Netw.* (TOSN) **2009**, *5*, 20-32.
27. Engel, J.M.; Zhao, L.; Fan, Z.; Chen, J.; Liu, C. Smart Brick: A Low Cost, Modular Wireless Sensor for Civil Structure Monitoring. In *Proceedings of the International Conference on Computing, Communications and Control Technologies (CCCT)*, Austin, TX, USA, August 2004.
28. Kim, S.; Pakzad, S.; Culler, D.; Demmel, J.; Fenves, G.; Glaser, S.; Turon, M. Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*, Cambridge, MA, USA, April 2007; pp. 254-263.
29. Szewczyk, R.; Mainwaring, A.; Polastre, J.; Culler, D. An Analysis of A Large Scale Habitat Monitoring Application. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Baltimore, MD, USA, November 2004; pp. 214-226.
30. Nasipuri, A.; Cox, R.; Alasti, H.; Van der Zel, Luke; Rodriguez, B.; McKosky, R.; Graziano, J. Wireless Sensor Network for Substation Monitoring: Design and Deployment. In *Proceedings of the 6td ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Raleigh, NC, USA, November 2008; pp. 365-366.
31. Liang, M.; Liu, J.; Luo, L.; Terzis, A. *RACNet: Reliable ACquisition Network for High-Fidelity Data Center Sensing, Technical Report, MSR-TR-2008-144*, Microsoft Research: Redmond, WA, USA, 2008.

© 2010 by the authors; licensee Molecular Diversity Preservation International, Basel, Switzerland. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).